COMP6481 Solving Problems Assignment Part B

sw734@kent.ac.uk

April 7, 2025

1 Introduction

After using the given data frames to come up with a constructive response to why and how the University's environment score could be improved, I can confirm there are some areas that have room for improvement, for example staff contribution and an increase in doctorates. The suggestions I am providing for increasing the universities environment score have come from comparing the REF 2021 data groups with these being universities with a better environment score and universities with a worse environment score

The aim of providing evidence to support my suggestions is so the university can prepare for the next REF assessment and ultimately increase their environment score, I believe that the suggestions that follow will lead to a positive impact to the environment score for Kent.

2 Improvement Suggestions

As I have already mentioned, from my interpretation of the data and text provided by these data frames I believe there are a few parts where Kent can improve upon for a better environment score. The improvements range from quantitative data such as pass rates, to qualitative data such as word frequencies within their unit or institutional environment statements. Here are the suggestions I have to help achieve a higher environment score.

2.1 Eligible Staff Submittion Rate

For my first suggestion, I decided to focus onto the `resultsSummary` data frame. This gave me a summary of all data from universities with a better or worse environment score than Kent. With this data frame I focused on data with the column `TYPE` being `ENV WORSE THAN KENT` ascending, as well as focusing on data with the column `TYPE` being `ENV BETTER THAN KENT` descending. This gave me the information on the top performing universities and the best universities which had a lower environment score than Kent. Now, when looking at how many eligible staff submitted as well as looking at how well they did, there is a clear link between how well a universities environment score was and how many staff submitted with a high score.

For example, the top performing university in `ENV WORSE THAN KENT` was `Birkbeck College`, with `100%` of eligible staff submitted with a 50/50 split of 2- and 3-star submissions (with the highest possible being 4 stars). The top performing university in the `ENV BETTER THAN KENT` was `The University of Manchester`, `100%` of eligible staff submitted with all of them achieving 4-star submissions. From the data I have got from my python snippets, I can infer and suggest that Kent make sure that every eligible staff

member submit as well as making sure they have had the correct guidance and education to achieve the highest rating possible as data shows these two have a close relation with a better environment score.

Another point to add is when focusing onto the `FTE` columns you can see that there is a lower `FTE` of submitted staff in universities with a lower environment score and the `FTE` is a lot higher universities with a better environment score when comparing to Kent. `FTE` stands for Full Time Equivalent, which kind of means that the higher `FTE` the more workload the university handles which infers more staff. The highest scoring university (Manchester) had plus `18.40` over Kent's `FTE` score and the top performing university in the lower than data had negative `9.1` under Kent's `FTE` score (Birkbeck College). This data leads me to the suggestion that Kent should increase their `FTE`, this ultimately means a higher workload and more staff which links to a higher environment score.

2.2 Number Of Doctorates From 2013 To 2019

Another suggestion I can give for increasing the environment score is increasing the number of doctorates per year, after looking at the data provided in the docAwards data frame I can see that the higher the annual number of doctorates the higher the environment score. The way I manipulated the data to easily see this relation between them is by getting the mean number of doctorates per year for each university and adding it to a new column in the data frame being `Mean number of doctoral degrees`.

When looking at `Manchester` and `Birkbeck College` again it further suggests that this is a good suggestion, in the `ENV BETTER THAN KENT`, Manchester is second place this time with a mean annual number of doctorates being `32.4`. The top university this time was `University of Southampton` with an annual number of `34.4`, Southampton was third best within my first suggestion. Within `ENV WORSE THAN KENT`, Birbeck has fallen down to eighth best university with an annual number of doctorates being `3.2`. After looking at the data I have generated from the data frame, the fact that Birbeck has fallen behind due to a lower number of doctorates and Manchester has near enough stayed the same due to having a high number of doctorates annually tells me that doctorates play a big part of a great environment score. This leads me to believe that suggesting Kent to increase the number of doctorates annually would most definitely increase the environment score.

2.3 Number Of Distinct Tokens In Unit Environment Statements

Now, focusing on the text you can see a link between the number of unique tokens and top performing universities. After manipulating the text provided in `unit_df`, I

can see that the top universities (such as Manchester) have a greater number of unique tokens in their unit environment statements when compared to the best universities who performed worse than Kent (such as Liverpool). The `unit_df` holds the unit environment statement of each university. This is why I am going to make the suggestion for Kent that they increase the number of unique tokens within their unit statement. The way I got all unique tokens was firstly I made all tokens lowercase before processing the unique amount of tokens, as this would inflate the number without it making it inaccurate, then I got the length of the data set holding the unique tokens. The results I got are as follows.

Manchester's distinct token count was `3059`, while Liverpool's was `2235`, the reason I am not using Birbeck College is because they do not have a unit statement which further supports the need of one as well one with distinct words. Once I had the text processed like this, I decided to go a step further to further validate my suggestions as well as the evidence I am giving to support it. My next step was to take out stop words as well as punctuation and add it to a new column in the data frame called `distinct no stopwords or punctuation`. When recounting the stop words without punctuation or stop words, the results maintain their purpose, that being showing that the universities with lower distinct words are performing worse in the environment score. Whether this distinct word count is lower due to an increase of stopwords or just a decrease in distinct words, either way the point still stands. University of Manchester's distinct word count is now `2188` while Liverpool's is now `1800`. After looking at the text, my suggestion still stands for increasing unique words within the unit statement while keeping stop words to a minimum.

3 Conclusion

To conclude my findings and suggestions for Kent, when compared to other universities Kent has overall done well to get a good environment score. But there is room for improvement. The suggestions I mentioned are a good place to start, those being all staff submitting as well as aiming to achieve a high score, increasing the number of doctorates given out on an annual basis and increasing the number of distinct tokens found within the unit environment statement. These all will help increase the environment score.

Furthermore, there are a few more processing techniques that could be used to suggest other actions that could increase the environment score, for example for the text, speech tagging can be used to see what kind of words are used and to see which kinds of words are used in relation to a high environment score, for example a high use of nouns or adjectives may be used in universities with a higher environment score. For the data, looking into the annual income as well as where the income comes from may show a link

towards a higher environment score, this can be found within the `researchIncome` data frame.

So in the end, these are the improvements I am suggesting with the aim of improving the environment score, I believe the evidence I have provided supports that these will indeed improve the environment score as well as the extra improvements I have suggested.

A Appendix: Code Snippets

A.1 Code Snippet For: Eligible Staff Submittion Rate

```
# result worse than kent ascending
results_worse_descending = resultsSummary[(resultsSummary
   ['TYPE'] == 'ENV WORSE THAN KENT') & (resultsSummary['
   Profile'] == 'Environment')].sort_values(by = '
   Environment +-', ascending=False)
results_worse_descending.head(100)

# results better than kent descending
results_better_descending = resultsSummary[(
   resultsSummary['TYPE'] == 'ENV BETTER THAN KENT') & (
   resultsSummary['Profile'] == 'Environment')].
   sort_values(by = 'Environment +-', ascending=False)
results_better_descending.head(100)
```

Listing 1: Best Environment Scores For Worse And Better Than Kent

A.2 Code Snippet For: Number Of Doctorates From 2013 To 2019

```
# get the mean number of doctirates from 2013 to 2019 for
    universities scoring better than kent
docAwards['Mean number of doctoral degrees'] =
    docAwards[['Number of doctoral degrees awarded in
       academic year 2013', 'Number of doctoral degrees
       awarded in academic year 2014',
               'Number of doctoral degrees awarded in
                  academic year 2015', 'Number of
                  doctoral degrees awarded in academic
                  year 2016',
```

```
                'Number of doctoral degrees awarded in
                   academic year 2017', 'Number of
                   doctoral degrees awarded in academic
                   year 2018',
                'Number of doctoral degrees awarded in
                   academic year 2019']].mean(axis=1)

docAwards_better_descending = docAwards[docAwards['Sample
   '] == 'ENV BETTER THAN KENT'].sort_values(by='Mean
   number of doctoral degrees', ascending=False)
docAwards_better_descending.head(100)

# get the mean number of doctirates from 2013 to 2019 for
    universities scoring worse than kent
docAwards['Mean number of doctoral degrees'] =
    docAwards[['Number of doctoral degrees awarded in
       academic year 2013', 'Number of doctoral degrees
       awarded in academic year 2014',
                'Number of doctoral degrees awarded in
                   academic year 2015', 'Number of
                   doctoral degrees awarded in academic
                   year 2016',
                'Number of doctoral degrees awarded in
                   academic year 2017', 'Number of
                   doctoral degrees awarded in academic
                   year 2018',
                'Number of doctoral degrees awarded in
                   academic year 2019']].mean(axis=1)

docAwards_better_descending = docAwards[docAwards['Sample
   '] == 'ENV WORSE THAN KENT'].sort_values(by='Mean
   number of doctoral degrees', ascending=False)
docAwards_better_descending.head(100)
```

Listing 2: Mean Number Of Doctorates From 2013 To 2019 For Worse And Better Than Kent Per Year

A.3 Code Snippet For: Number Of Distinct Tokens In Unit Environment Statements

```
# how many distinct tokens are there?
unit_df['distinct tokens'] = unit_df['new'].apply(set).
   apply(len)
# unit_df.head()
# The University Of Manchester
unit_df[unit_df.index == 'The University of Manchester']
# Birkbeck College or Liverpool John Moores University
unit_df[unit_df.index == 'Liverpool John Moores
   University']
```

Listing 3: Number Of Distinct Tokens In Unit Statement

```
# Remove all stop words and punctuation from the tokens
from nltk.corpus import stopwords
unit_df['no stopwords no punctuation'] = unit_df['new'].
   apply(lambda remove : [word for word in remove if word
   .isalpha() and word not in stopwords.words('english')
   ])
# count distinct tokens again
unit_df['distinct no stopwords or punctuation'] = unit_df
   ['no stopwords no punctuation'].apply(set).apply(len)
# unit_df.head()
unit_df[unit_df.index == 'The University of Manchester']
unit_df[unit_df.index == 'Liverpool John Moores
   University']
```

Listing 4: Number Of Distinct Tokens In Unit Statement Without Punctuation And Stop Words